

A single-system interface using the IBM 3270-PC

by M. M. Ghiotti

Many businesses use a variety of terminal types connected to central host computers. Presented here is a rationale and the experience gained with a single terminal type—the IBM 3270-PC—interconnected with hosts via the Application Program Interface to achieve enhanced user efficiency.

The 1970s were noted for the development of complex host-based batch and on-line applications. This was also the time when stand-alone word processing applications were coming into general use. Both kinds of applications were designed to meet the needs of changing business. The goal of both applications was the same: to provide timely and accurate information upon which each end user could rely. Although individual applications shared this common goal, the software and hardware required were often quite diverse.

The end-user interfaces for accessing these applications were generally fixed-function devices or workstations. A workstation was a configuration of input/output equipment—usually a keyboard and a display—engineered to automate a specific function. Workstation selection was dictated by the application needs. If the function to be automated involved primarily word processing, a stand-alone device with no host connection was used. Such a workstation often consisted of a uniquely designed keyboard and a monochrome display. Inquiry applications generally required only a monochromatic display and a simple keyboard that were connected to a host computer system to execute the application. If the application involved graphics, a color display with program symbol support and a somewhat more complex keyboard were provided.

The result was often the use of a variety of workstations within the same department. A multiplicity of workstations tended to be inefficient to the degree that an end user might be required to develop a separate set of skills to use each type of workstation.

This situation was complicated by the introduction of low-cost microprocessors. Many end users, especially those whose tasks required data from a host computer, were delayed by terminal response time and the lead time required by the data processing department to adapt existing applications to the new ways of processing data. As a result, business executives and professionals began to see stand-alone microprocessor workstations as a way of going around obstacles to productivity that had been induced by the system. With microprocessors, end users could develop or buy application software to do their problem-solving and decision-support activities. This approach enabled end users not only to bypass an application backlog queue, but also to avoid terminal response-time delays.

Obviously, much of the data needed as input for these microprocessor workstations was at the same time resident on the host computer system. Having access to these host-based data would eliminate duplication and promote consistency of the data if they were readily available. This need for host computer

© Copyright 1985 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

access suggested the combining of some fixed-function workstation characteristics with those of the independent workstation used for local processing. A workstation that could provide a combination of host-computer access and local processing appeared to be the configuration sought after. Such a workstation, operated within the security requirements of an enterprise, could provide graphics support, word processing, and access to data on any host computer system within the company. The major prerequisite remained that the workstation user not be required to develop data processing skills to use the workstation or to access host-computer-based data.

In view of this history of workstations in business applications, the approach of using specific application requirements as the sole selection criteria for workstations is no longer necessary. The solution discussed in this paper is that of providing a single workstation or interface through which an end user can easily access the data and applications needed for his tasks. The workstation discussed also provides a means for combining data from all the sources, so that the user can perform his tasks in a sequence dictated by the application rather than by the equipment.

Criteria for a cooperative processing environment

Underlying any discussion of applications of computers, whether long and heavy scientific computations or rapid-fire, short business computations, are requirements for accuracy, timeliness, cost-effectiveness, and ease of use. The following are considerations of these requirements as they apply to workstations.

Accuracy is a quality held by that which is free of error.¹ The use of microprocessors with data-file storage capacity introduces the ability for end users to create and maintain local data bases, and the potential for making errors—inaccuracy. Frequently, the data are keyed into the file from a listing obtained from the host data base. The possibility that the listing is not a current reflection of the host file and the potential for making errors during the keying process affect the accuracy of those data. In any enterprise, there may be many locally maintained files upon which business decisions are being made. Thus we must provide the end users with a method of ensuring that their decisions are being based on common and accurate data. So that the end user starts with common and accurate data, the

host is the central repository for corporate data, and the workstation is the means by which each individual accesses needed data. Providing access to the host-based data not only ensures that all users are basing their decisions on the same set of data, but also that the data reflect all updates, so as to reflect the current business environment.

Timeliness is one of the major attractions of microprocessors in the business community. One factor is the ability to perform specialized local applications to enhance the decision-making process. Timeliness is the ability to perform these applications very quickly in such a way that they are not dependent

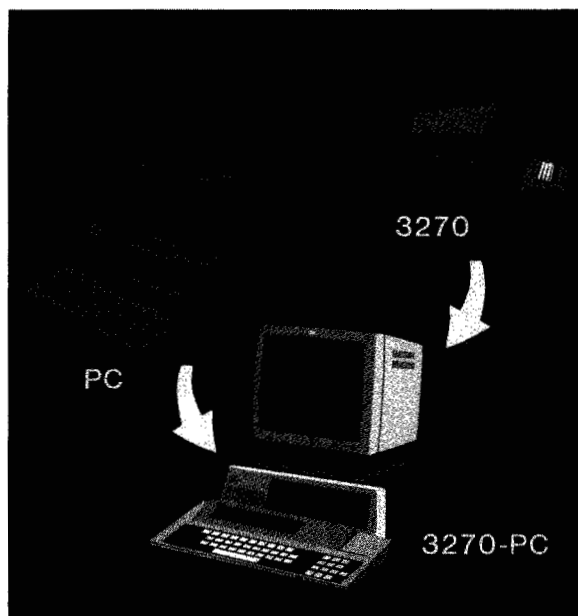
**With modern microcomputers, users
perform applications simply and
easily.**

on terminal response time. Microprocessors not only provide the user with the capability of responding to a specialized request very quickly, but they also allow the user to interject other factors or new variables that might anticipate effects that business trends, for example, might have on decisions.

It may not be cost-effective to select workstations solely on the basis of each application need because this can result in both fixed-function and programmable workstations in the same department. Across the department, this may be an uneconomical approach to automating a function because it introduces multiple interfaces or workstation types that may detract from end-user productivity. Furthermore, each interface or workstation may require a separate connection to the host system, which increases the number of communication lines and supporting hardware and software. A single workstation type that is capable of addressing all the needs of an end user can lead to a sizable cost savings to the enterprise.

Our experience has been that there is a definite requirement for end users to access host systems and

Figure 1 The 3270-PC combines the function of the IBM Personal Computer with the capabilities of the IBM 3270 display products



to do so with ease. There have been times when host systems were not considered easy to use. This was not always important because there was usually very little direct interaction between hosts and the end-user community. Systems designers and application programmers shielded the user from the complexities involved in using the host. Now, however, many microcomputer systems are designed for use by persons other than data processing professionals. With modern microcomputers, users perform applications simply and easily, concentrating on the desired results rather than on the intricacies of the machines. Little or no data processing knowledge is required. Even so, the need to access host data bases to provide timely and accurate results introduces a degree of complexity that does not exist in the stand-alone environment. Thus, ideally, an all-purpose professional workstation provides a user the ability to easily access, display, and/or view data from multiple host computer applications concurrently, with the ability to perform applications locally.

These, then, are the criteria: accuracy, timeliness, cost-effectiveness, and ease of use. The challenge is to provide a single workstation or interface that meets these criteria within the security structure of the organization. The workstation is to be used in such modes as host data processing, local data proc-

essing, document processing, graphics, and message transmission and reception. The workstation must also provide a means for combining data from multiple sources so that the user can perform the required tasks in a sequence dictated by the need, rather than by the equipment.

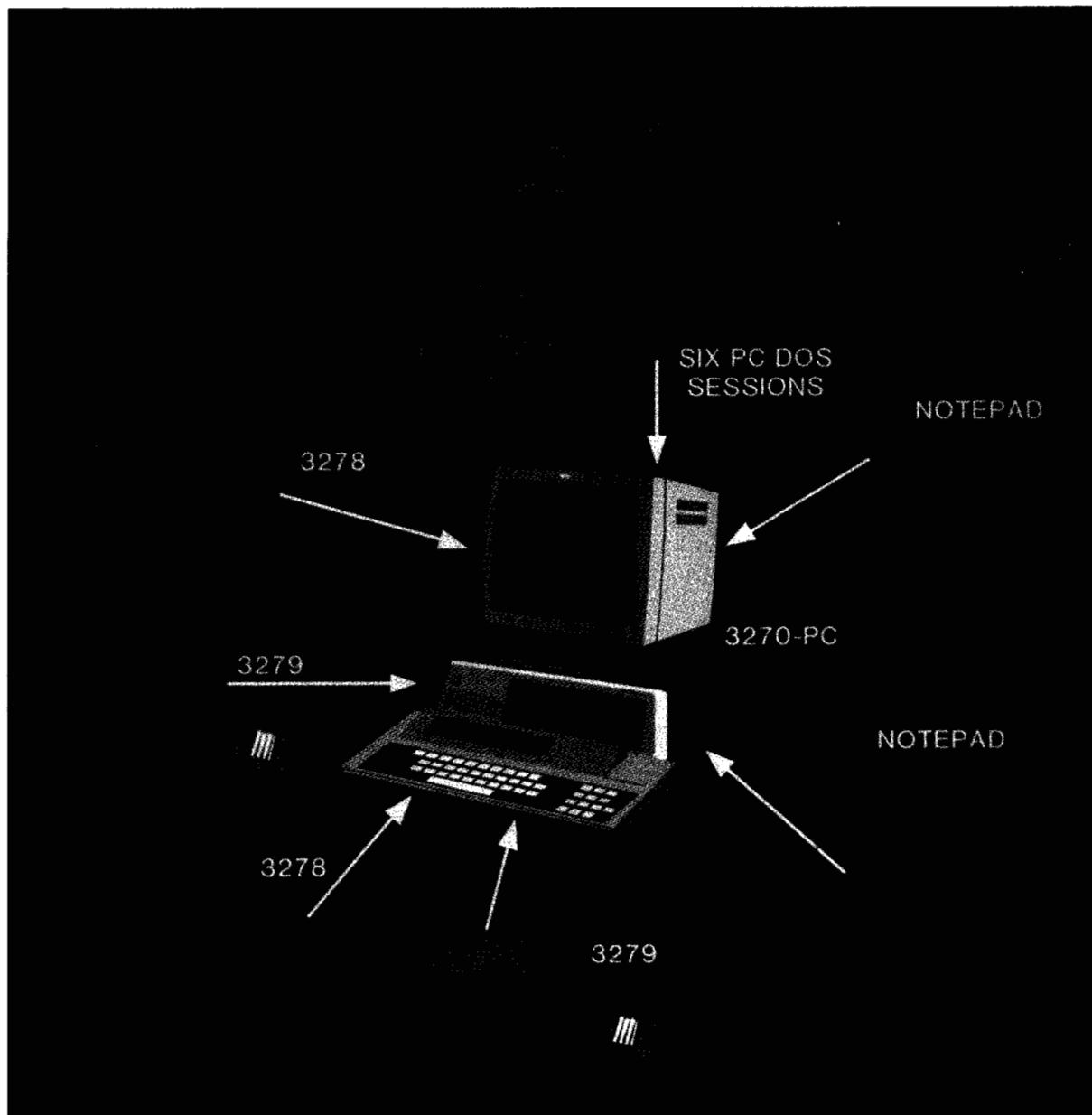
Design of the cooperative processing environment

The IBM 3270-PC and the 3270-PC Application Programming Interface (API) provide end users with a single interface combining both 3270 terminal-host processing and local personal computing. At the same time, they provide a facility to allow complete programmed freedom to communicate data between the two modes of operation. This system design not only satisfies the need for a programmable workstation with 3270 host affinity in a single hardware device, but also provides an environment for cooperative processing. Host computer application programs and workstation application programs can thus be designed that will complement and interact with one another.

Since the 3270-PC and the API are relatively recent developments, there is not a general awareness of their capabilities. Therefore, this paper first examines each device to show that together they are the appropriate combination to eliminate the problem of multiple workstation types. In my work to date, to demonstrate this solution to end users, I have used a program offering called the High-Level Language Interface, which allows a programmer to write an API program in a high-level language (e.g., Basic or COBOL) rather than writing in assembler language. The examples that follow the discussion of the 3270-PC and API are demonstration API programs that I have written in Basic and have used as vehicles to demonstrate the concepts of cooperative processing and a single system interface. [Throughout the remainder of this paper, reference to "programs" means these demonstration programs, which run under Personal Computer Disk Operating System (PC DOS) and make use of the 3270-PC API as API programs.] Time savings or productivity improvement figures mentioned here are estimates that those attending the demonstrations believed they could realize in their particular environments.

The IBM 3270-PC. The IBM 3270-PC shown in Figure 1 is architecturally appropriate for providing a single, all-purpose interface or workstation. It can be used as a fixed-function 3270-type device, providing the

Figure 2 Customization of the 3270-PC control program allows tailoring of the workstation operating environment to meet individual requirements

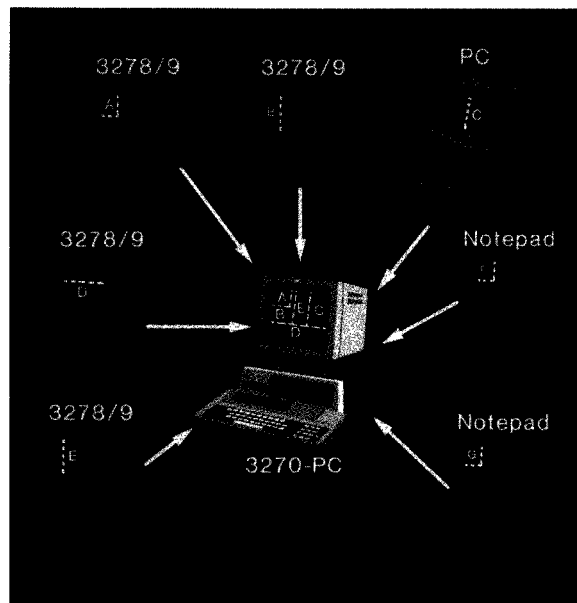


user with the ability to perform host-based text, data, and graphics applications. It is also an integrated IBM Personal Computer (IBM PC), and therefore it is capable of processing graphics applications and local, stand-alone text. A single keystroke is all that is required to alternate between these two modes of operation.

The functions of the 3270-PC and the 3270-PC Control Program that address the single interface requirements as outlined above are the following:

Customization. A customization feature, illustrated in Figure 2, allows the workstation to be tailored to meet the requirements of each end user. During the

Figure 3 Windows on a single presentation space can be used to view an area of each session



customization process,² the workstation can be defined to utilize the following:

- Up to four host-interactive 3270 sessions
- Up to six IBM PC DOS³ sessions
- Two electronic notepads

Each 3270-type session or application screen is equivalent in function to a separate 3178, 3278, or 3279 fixed-function terminal and a unique operating session. Each session can represent applications from the same or different host computer systems. Similarly, each IBM PC session that is defined looks and works as though it were a stand-alone PC DOS application. The notepads are functionally equivalent to a one-page scratch pad for jotting down information that can be saved or discarded.

Although the ability to utilize all these advanced functions exists, it is not necessary to do so. A given application may require only two 3270-type host sessions, a single IBM PC session, and one notepad. The *customization process* (somewhat similar to a SYSGEN for a host computer) allows for the tailoring of the IBM 3270-PC control program to meet the needs of the application. Therefore, only those sessions that are required need be defined.

Screen management. The advanced screen management capabilities of the 3270-PC control program

permit a user to view multiple sessions simultaneously through windows that can be located anywhere on the physical screen (i.e., the presentation space). This is shown in Figure 3. A window can be defined as the entire presentation space, or it can be as small as one character. The important aspect of windowing is that it allows for the concurrent viewing of portions of multiple presentation spaces on a single presentation space. These application windows can be sized (i.e., made larger or smaller) and moved or positioned on the screen, as the application dictates.

Copy function. A user can copy host data from an active 3270-type host session to any other 3270-type session or to a notepad session. By another copy method, the user can copy from a Personal Computer session to a 3270-type session or notepad and save the copied information in a file for later use. This function provides an interactive mechanism to access selected portions of host data. The copy function is illustrated in Figure 4.

File transfer. In addition to the copy function, entire data files, either text or programs, can be transferred to or from an MVS/TSO or VM/SP host system during an IBM Personal Computer session. All conversion from ASCII to EBCDIC is handled by the 3270-PC Control Program, as shown in Figure 5.

Using these functions, the user has complete flexibility with respect to the movement of data to and from all sessions—host or local—and the manner in which the data are displayed. Thus, the capabilities exist to achieve a true cooperative processing environment. With this workstation, a user can have a single hardware interface (i.e., one display and keyboard) through which to gain access to pertinent data from various sources and integrate those data into local processing applications.

Thus far, we have shown that the 3270-PC can provide a single hardware interface, thus simplifying the actual interface for the user. What remains to be simplified or removed entirely is the requirement that the end user have knowledge of the data processing environment to be interacted with. Without the API, before a user can use the workstation as a tool to gain access to the data required to perform the actual task, knowledge of the following must be acquired:

- Personal Computer disk drive designators and file specifications
- Host and local session names and how to specify them

Figure 4 Data can be copied from a host, PC, or notepad window into another host, PC, or notepad window

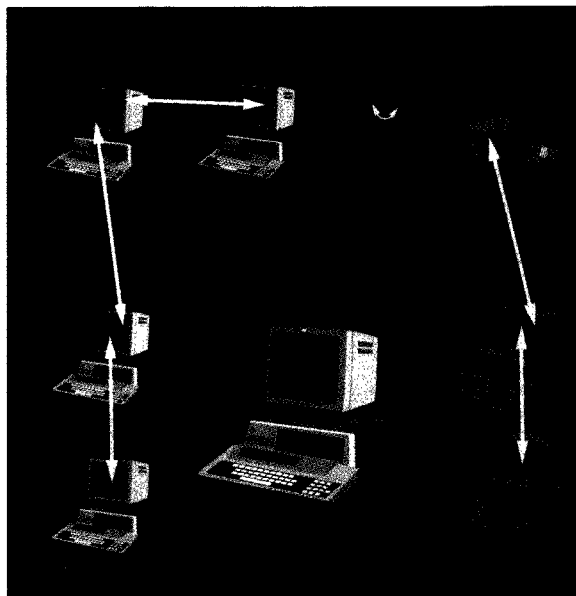
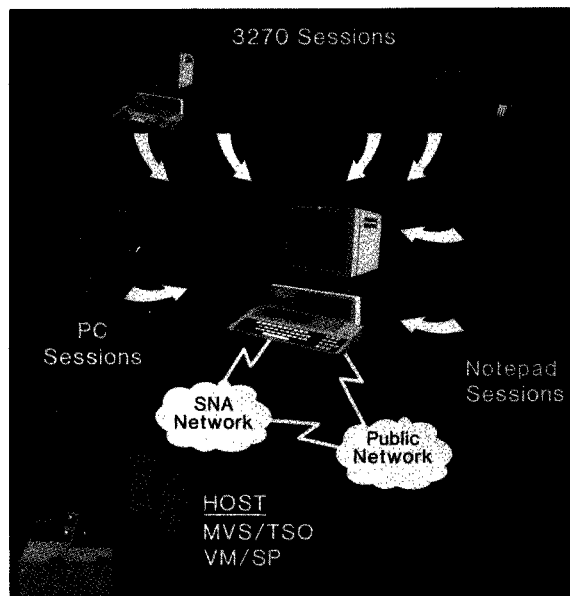


Figure 5 Files can be transferred between an MVS/TSO or VM/SP host system during a PC session



- The meaning of such data processing terms as append, CRLF, ASCII, LRECL, and BLKSIZE, and how each of these relate to transferring the required data to a Personal Computer disk file
- The operational characteristics of a 122-key keyboard that provides the user interface to the extensive functions of the 3270-PC
- Where, within the enterprise network, the required data reside and the procedures required to gain access to them

All of these are functions related directly to the operational characteristics of the Personal Computer when interacting with a host computer or to the methods that are required to access the host systems. Since none of these relate directly to the task at hand, it would be useful to have the ability to program these operations and have the Personal Computer perform them for the end user. The effect of such a procedure is to customize the 3270-PC workstation to the requirements of the end user. This leads to a discussion of the Application Program Interface (API) and how it can facilitate the development of a personalized or customized interface that is based on the requirements of a given end user.

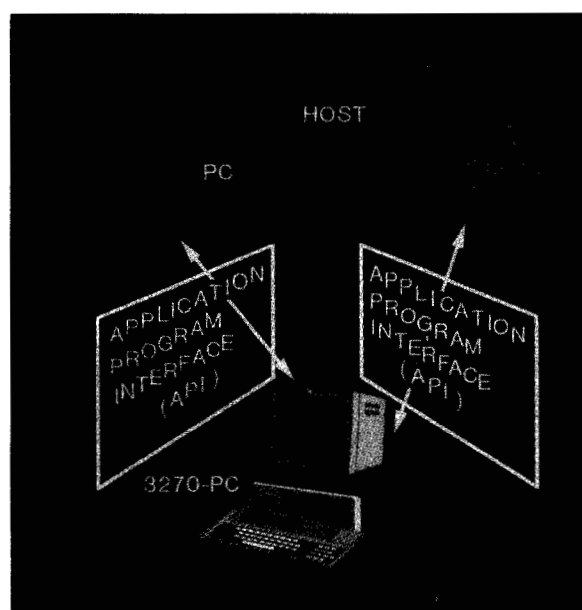
Application Program Interface

The user interface is one of the most important parts of a workstation. Without an easy-to-use interface

the user may find a function too complex and thereby defeat the purpose of the workstation. The Application Program Interface (API) of the 3270-PC control program is a tool that can be used to remove both the data processing complexities and the operational complexities of the 3270-PC workstation and make possible a single easy-to-use interface. API is the tool that provides the potential for achieving a greater degree of cooperative processing and tailors the workstation to the specific needs of each particular end user.

Thus far, we have seen that the 3270-PC allows access to many sources of data from a single workstation. With the range of functions of the 3270-PC, an end user can log on to a single application or multiple host applications on the same or different host computers from one workstation, rather than relying on the availability of multiple workstations. Host applications or sessions for the 3270-PC are a combination of data transfers and transactional processing. Between the host and local sessions, a high degree of flexibility in the movement of data to and from the different types of sessions is necessary to achieve full integration of host and local processing. When this integration of host and local processing is accomplished, cooperative processing results, giving the user the power and full use of the enterprise's resources.

Figure 6 The Application Program Interface provides a bridge between a local PC application program and the 3270-PC control program



The 3270-PC control program services that are available to the end user through the keyboard allow for this integration or cooperative processing between the host and local sessions. Associated with this function and flexibility, however, there is a degree of operational complexity. Without programming support, the end user has to design the method or manual procedures required to achieve the necessary cooperation, and must understand how to invoke the control program services to accomplish this.

The Application Program Interface (API) is an interface between an application program in a local Personal Computer session and the 3270-PC control program. Just as a bridge provides access to separate pieces of land, the API provides a bridge or interface between the local Personal Computer and the 3270-PC control program. (See Figure 6.) The API provides the facilities for intersession communications, allowing sessions to pass keystrokes, data, and status information between presentation spaces and programs, as shown in Figure 7.⁴ An API program can invoke the services of the control program to perform the following functions:

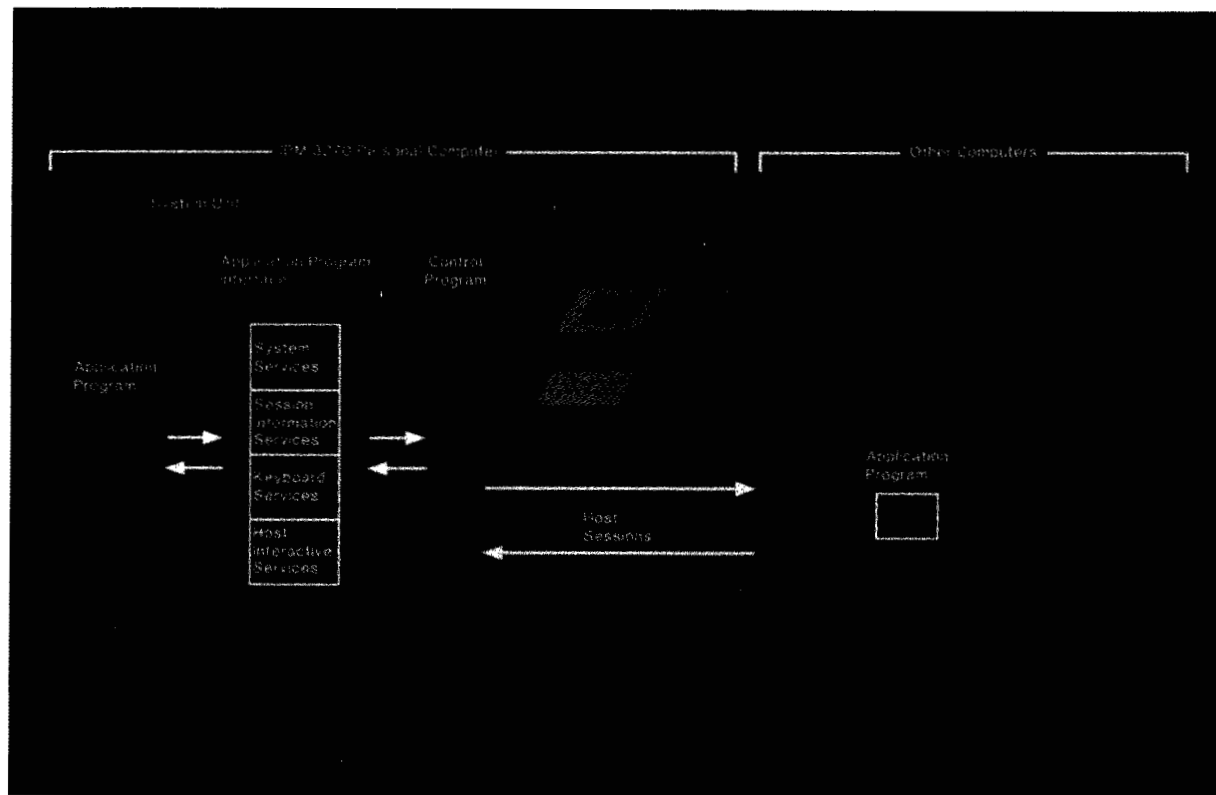
- Read/write keystrokes, allowing a Personal Computer session to send keystrokes to other sessions or receive keystrokes from sessions

- Lock the keyboard to prevent operator input when keystrokes are being sent from the Personal Computer program to a session
- Obtain status information for any session to determine the active sessions or to establish communication parameters
- Monitor host events or status for such functions as event completion, line disconnect, and reconnect
- Invoke the save/restore utilities of the control program to save or restore to or from disk or diskette customized screen definitions and notepad data
- Invoke the file transfer facility to send or receive data files to or from a host session
- Copy data (entire presentation spaces or specific fields) to or from active sessions

These are some of the functions of the API available to the Personal Computer application developer. Each individual function performs a piece of the task to be done; when combined, the individual functions constitute a powerful set of functions that can manipulate the application environment for the end user. These functions facilitate the coordination and integration of 3270-type host sessions and data and Personal Computer capabilities when an application is being implemented. The functions provide the facility to interact with host sessions to enhance the usability of the 3270-PC by making complex functions transparent to the end user. They can be employed to automate, customize, and simplify many functions that were previously available only through manual procedures executed by an end user. The examples given later in this paper illustrate this capability.

As mentioned previously, an API program is able to enter any and all of the workstation keys, thereby controlling the workstation and its interaction with the host sessions through programmed logic. The ability to request the services of the control program through programmed logic within an application actually controls what the workstation will do for the end user. All the data processing knowledge that an end user must acquire prior to actually performing his task, as well as the knowledge of the operational characteristics of the workstation, is no longer a requirement. All an end user requires is a knowledge of the data requirements of the actual task. The implementation of an application that will provide those data can be done by a programmer. Thus we see that the "programmed operator" becomes a reality.

Figure 7 The Application Program Interface provides for the flow of application program requests for services from the 3270-PC control program

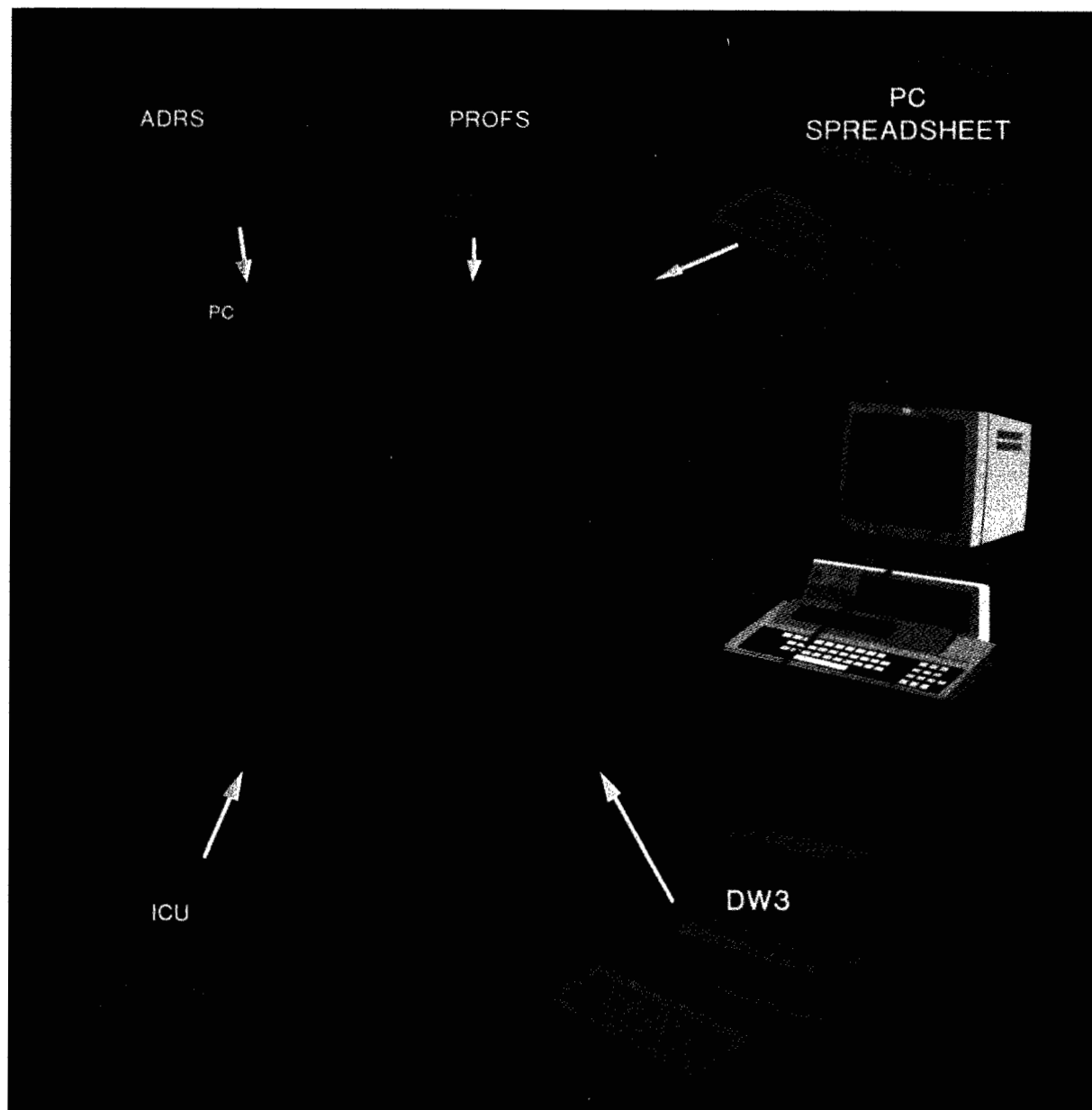


Along with removing operational complexity, the API program can be designed to gather data from multiple host sessions into one composite screen or presentation space and present the data to the end user without windowing. Data from the same or different hosts can be presented to the user in a format suited to each particular need. To achieve this, no changes need be made to the host computer applications. The inclusion of this type of function within the application removes the end user one step farther from the concerns of the host computer environment. Although two or more windows placed appropriately within the presentation space can be used to display these results, the composite screen is an even more friendly presentation. The important aspect of multiple sessions is not the windowing, but rather the capability of using multiple host sessions as a means of gathering information. All direct ties to host computer applications become completely transparent to the user, as shown by the examples that follow.

Recalling that cooperative processing is an application approach in which productive work is done partly by the host and partly by the workstation, it is obvious that the 3270-PC in conjunction with the 3270-PC API is well suited to dividing work between the 3270-PC and one or more host sessions. The 3270-PC and the API make it possible to split application workload as desired between host and workstation. The API program can build host interactive transactions in the workstation, send them to a host, and receive the responses back at the workstation. The application designer could use a combination of existing host transactions to create enhanced applications without modification to host code. The result is cooperative processing between the host computer and the workstation, without end-user intervention.

With access to multiple host sessions and the ability to send data to them, the API program becomes the system interface between an end user and the many functions of the 3270-PC. The 3270-PC workstation,

Figure 8 Programmed operator functions are exemplified by an application designed to implement the programmed operator concept



while executing the API program, presents a single system interface to the end user. Also, because there is the ability to customize the application to meet the data needs of a given end user, the 3270-PC may be considered to be an all-purpose professional workstation.

Examples

We have discussed how the use of an API program enables the customization of the 3270-PC workstation to yield the following:

- Programmed-operator functions

- Composite screen displays
- Cooperative processing between host(s) and workstation

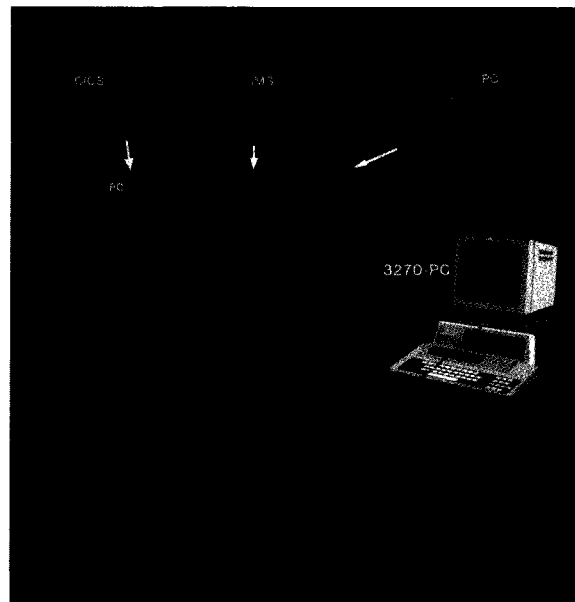
The following examples are actual API programs I have written that address each of these environments.

Programmed-operator functions are exemplified by an application designed to implement the programmed-operator concept. This example is illustrated by Figure 8. The interested users were the business professionals of a pharmaceutical organization. They required access to three host sessions. One session was the source of their electronic mail, their calendaring, and their scheduling facility. During the course of the day, they also required access to a host-based charting/graphing utility, Interactive Chart Utility (ICU) and the host-based reporting facility, A Departmental Reporting System (ADRS). Some users also wanted access to other host-based facilities (e.g., the Query Management Facility QMF and The Information Facility TIF). Only minor modifications to the base API program were made to tailor the application to their special, individual requirements.

In addition to these host-based tools, the users also needed to transfer data to and from a host and to utilize the local processing power of the Personal Computer for spreadsheet and word processing applications. One can clearly picture the huge amount of manual intervention required to perform these functions without some type of programming support.

With an API program, however, all of the complexity was removed. Program function key 1 (for logging on to host systems) enters the required sign-on signal and password when prompted to do so. The application not only establishes the host connections that are required, but also invokes the host-based application. Program function key 2 then connects the keyboard to PROFS, where it will remain until the user enters a particular code on the screen that signals the application to return to the main selection screen, as shown in Figure 8. The remaining host tools are operated similarly. Program function keys 5 and 6 build the required file transfer commands, using installation-established file names, and these keys also initiate the actual transfer. Finally, selection of program function key 7 or 8 causes the local application program to be loaded and transfers control to it. Upon completion of any of these events, the Personal Computer presentation space returns to the

Figure 9 An insurance company application illustrates composite screen displays

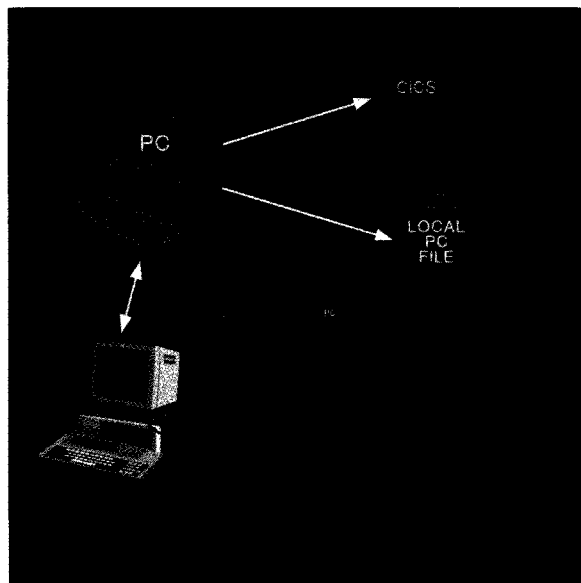


main selection screen. This demonstration showed the users that they could access the needed tools without operational complexities and without having to gain additional data processing knowledge. The operation of the workstation has been programmed into an application for a group of seventeen business professionals.

Composite screen displays are exemplified by an application that was designed for an insurance company (see Figure 9). When policyholder service representatives in the insurance firm receive an inquiry as to the status of a recent claim, it is necessary to log on to two separate host applications to obtain the data needed to respond to the inquiry. Data from the first application was formerly written down on paper, and the representative logged off that system. This was followed by a log-on to the second host application.

By the new system, the service representative enters the policy number on the Personal Computer screen. A transaction is formulated and sent to the host-based CICS customer information file to retrieve policyholder information. Concurrent with this, data are extracted from an IMS claims data base. The data from each host inquiry are then reformatted and displayed on the Personal Computer screen, operating as a composite screen. The service representative

Figure 10 An insurance company application illustrates cooperative processing



receives all the information required to respond to the inquiry and is relieved of such concerns as performing multiple host logons, initiating host transactions, and even making notes on paper. Some observers estimated that they could potentially realize a ten to fifteen percent productivity improvement in their environment by utilizing this type of application.

Figure 10 shows a simple but effective implementation of cooperative processing. Here, all policy information is retained centrally on the host, with agent-specific data being transferred to local Personal Computer data files on a regular basis. The local API application determines that the host has sent notification that the local files need to be updated. A file transfer request is then sent to the host, and the user is notified that the local files are being updated. The file is downloaded to the Personal Computer, translated into ASCII code, and stored on the local fixed disk. The user has been relieved of such concerns as file transfer syntax, data formats, and Personal Computer file names. Maintaining the policyholder records locally allows timely and accurate responses to inquiries made by an agent's policyholders. If a request is received for information on a policy not held locally, the API application initiates a host transaction to retrieve the information from the centrally maintained file. To the user, it is as though the query

has been satisfied locally, because the results are presented in the same format.

Although observers could not estimate accurately the time savings or productivity improvement, they felt that there certainly would be some. They did feel that such an application would lead to increased customer satisfaction, because agents would be able to respond to inquiries in a more timely manner.

These examples illustrate some of the capabilities of the 3270-PC API and the potential that it provides to tailor an environment, to add ease-of-use features to an application, and to develop a cooperative processing application. Each example includes a combination of host computer access and local Personal Computer processing designed to meet the needs of an end user through a single interface.

Some customers have begun to implement their own applications. One example is a very straightforward application that performs multiple logons for a systems development department. When the users power on their 3270-PCs, they enter one command followed by the required security information. They are then logged on to TSO/ISPF, and their second session goes through CICS to DISOSS to DISOSS/PS to Personal Manager to display their calendar to them. This particular customer has subsecond response time and still feels that this application saves at least eight minutes per user when each one powers on.

Another application that has been implemented is one in which multiple screens had to be viewed to determine whether they contained certain information. If so, the screen(s) were printed; these screen dumps constituted a report. With the API application, the operator now enters search criteria. The application searches each screen, extracts the data when a match is found, and generates a formatted report. The customer estimates a time savings of from twenty-five minutes to one and one-half hours per operator per day for each of fifteen operators.

Concluding remarks

There is a fair amount of Personal Computer programming required to implement this approach to providing a single-system interface. Involvement of Management Information Systems staffs is also vital in implementing these API programs. In addition, because they are very susceptible to changes that may be made to host systems operation, there is a definite need for coordination between the MIS staff

and the systems staff. Some methods that would simplify these requirements in the future are the following.

A common Personal Computer application program to generate a customized API program for each end user, based upon individual requirements, would greatly simplify the techniques discussed in this paper. This approach, however, would continue to place the burden of definition and maintenance on the end user.

A host-computer-based API application generator that would generate customized API programs and subsequently transfer them to the appropriate workstations would also be helpful. This would be administered by the central data processing facility, thereby relieving the burden on the end user and also providing a more centralized control facility. Since data processing personnel would be involved, they would be able to incorporate system changes when required and to distribute the revised API programs simultaneously with the implementation of host-computer-system changes. This would help to eliminate workstation downtime and the need for the end user to regenerate API programs.

The ability for the host computer to initiate the transfer of program and data files to a workstation would not only be an aid to the orderly and controlled distribution of data and programs, but would also address the requirement that the workstation have access to accurate data. Distribution could be controlled by such means as date and time stamps and host-based lists of authorized end users.

In summary, the use of the 3270-PC and the API can result in numerous benefits to the end user as well as to an enterprise. Use of this workstation can do the following:

- Provide a single system interface through which an end user can access data and applications, regardless of where they might reside
- Help eliminate repetitive, non-task-related procedures, resulting in increased end-user productivity
- Reduce cost associated with automating a task to the extent that the purchase of another application-specific workstation may not be required
- Take advantage of the large investment most companies have in data and applications available on hosts, through cooperative processing and integration with local applications

- Reduce line cost, because fewer workstation types are required
- Offload the host by performing new or additional application processing at the workstation level
- Provide the capability of developing an enhancement to a host-based application without modification to the host software

Cooperative processing, implemented through the API and the 3270-PC, provides the function plus flexibility that allows systems integration of host and workstation in today's environment.

Cited references

1. *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699, IBM Corporation; available through IBM branch offices.
2. *IBM Personal Computer Control Program User's Guide and Reference*, SC23-0103-3, IBM Corporation; available through IBM branch offices.
3. *IBM Personal Computer Disk Operating System, Version 2.1*, 6024-125, IBM Corporation; available through IBM branch offices.
4. *IBM 3270 Personal Computer Control Program Programming Guide*, SC23-0165-0, IBM Corporation; available through IBM branch offices.

General references

- B. C. Goldstein, A. R. Heller, F. H. Moss, and I. Wladawsky-Berger, "Directions in cooperative processing between workstations and hosts," *IBM Systems Journal* 23, No. 3, 236-244 (1984).
- S. Gray, "IBM's 3270-PC: Bridging the gap between micro and mainframe," *Interface Age* 9, No. 4, 104-109 (1984).
- L. Augustin, "The mainframe connection: IBM's 3270-PC," *BYTE* 9, No. 9, 231-237 (1984).
- S. Atre, "Spreading the wealth of DBMS," *Computerworld* 17, N50A (Software Buyer's Guide), 31-34 (December 14, 1983).

Micki M. Ghiotti IBM National Accounts Division, One IBM Plaza, 08/098, Chicago, Illinois 60611. Ms. Ghiotti is an advisory systems engineer in the Chicago Insurance branch office of the National Accounts Division. She joined IBM in 1965 as a systems programmer in the Poughkeepsie development laboratories. Ms. Ghiotti is currently involved as a workstation specialist concentrating on the 3270-PC and API. She was previously involved with a wide range of large and small systems hardware and software both within IBM and as an IBM customer. At one time, she had her own consulting business, in which she was involved mainly with PC and host-connectivity projects. Ms. Ghiotti has a B.A. degree from Ladycliff College, Highland Falls, NY, in languages and political science.

Reprint Order No. G321-5251.